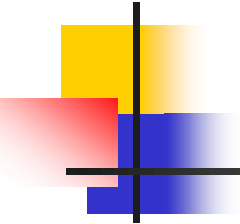




Programski prevodioci 1

Lekcija – Parsiranje od dna ka
vrhu, prvi deo

- 
-
- Uvod
 - SHIFT-REDUCE Parsiranje
 - Konstrukcija LR(0) parsera
 - Konflikti u SHIFT-REDUCE parsiranju




Pojam parsera

- **Parser:** algoritam koji utvrđuje, za zadatu ulaznu sekvencu, da li pripada jeziku zadate gramatike.
- Ako pripada, parser određuje izvođenje posmatrane ulazne sekvence iz startnog neterminala gramatike.



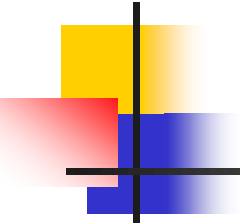
Strategije parsiranja

- **Od vrha ka dnu** (engl. **top-down**):
parser konstruiše stablo izvođenja
počev od korena ka listovima
- **Od dna ka vrhu** (engl. **bottom-up**):
parser konstruiše stablo izvođenja
počev od listova ka korenu



Potisni automat (pushdown automaton)

- Neformalno: potisni automat = konačni automat + stek
- PA su jednako "jak" formalizam za opis jezika kao bezkontekstne gramatike

- 
-
- Uvod
 - SHIFT-REDUCE Parsiranje
 - Konstrukcija LR(0) parsera
 - Konflikti u SHIFT-REDUCE parsiranju



BU parsiranje po principu potiskivanja i sažimanja (shift-reduce)

- Parser po principu potiskivanja i sažimanja (*engl. shift-reduce*) predstavlja potisni automat, koji je definisan:
 - skupom ulaznih simbola
 - skupom simbola steka
 - početnim stekom i
 - potisnom tabelom i kontrolnom tabelom.

Primer SHIFT-REDUCE parsera

1. $\langle S \rangle \rightarrow \langle A \rangle \langle A \rangle \langle S \rangle$

2. $\langle S \rangle \rightarrow \varepsilon$

3. $\langle S \rangle \rightarrow a$

4. $\langle A \rangle \rightarrow a$

	$\langle S \rangle$	$\langle A \rangle$	a
∇	$\langle S \rangle_0$	$\langle A \rangle_1$	a_x
$\langle S \rangle_0$			
$\langle A \rangle_{11}$		$\langle A \rangle_{12}$	a_4
a_x			
$\langle A \rangle_{12}$	$\langle S \rangle_1$	$\langle A \rangle_{11}$	a_x
a_4			
$\langle S \rangle_1$			

(a) potisna tabela

	a	$- $
∇	SHIFT	REDUCE (2)
$\langle S \rangle_0$	REJECT	ACCEPT
$\langle A \rangle_{11}$	SHIFT	REJECT
a_x	REDUCE (4)	REDUCE (3)
$\langle A \rangle_{12}$	SHIFT	REDUCE (2)
a_4	REDUCE (4)	REDUCE (4)
$\langle S \rangle_1$	REJECT	REDUCE (1)

(b) kontrolna tabela



Elementi SR parsera

- Skup ulaznih simbola parsera sastoji se od skupa terminalnih simbola zadate gramatike proširenog simbolom -|.



Elementi SR parsera(...)

- **Simboli parserskog steka** obeležavaju vrste potisne i kontrolne tabele.
 - svaki simbol steka odgovara nekom gramatičkom simbolu.
 - simboli steka kodiraju i dodatne informacije, o kojima će biti reči pri razmatranju konstrukcije parsera. Ove informacije su potrebne radi jednoznačnog određivanja parserskih aktivnosti;
 - zbog postojanja ovih dodatnih informacija, više različitih simbola steka odgovara jednom istom gramatičkom simbolu.



Elementi SR parsera(...)

- **Potisna tabela** određuje koji simbol steka se u određenom trenutku potiskuje na stek.
 - Redovi potisne tabele označeni su simbolima steka a kolone gramatičkim simbolima.
 - U svakom ulazu tabele nalazi se najviše jedan stek simbol (prazni ulazi označavaju grešku)



Elementi SR parsera(...)

- **Kontrolna tabela** određuje akciju parsera na osnovu tekućeg simbola na vrhu steka i tekućeg ulaznog simbola.
- Postoji četiri tipa **parserskih akcija**:
 - ACCEPT - uspešan završetak parsiranja;
 - REJECT - neuspešan završetak parsiranja.
 - SHIFT (potiskivanje):
 - PUSH(potisna_tabela[vrh_steka, tekući_ulaz])
 - ADVANCE
 - REDUCE(p)
 - POP x dužina desne strane p
 - PUSH(potisna_tabela[vrh_steka, leva strana p])
 - RETAIN



Operacije potisnog automata

- **Nad ulaznom sekvencom:**
- **ADVANCE** označava prelaz na sledeći ulazni simbol
- **RETAIN** označava da se tekući simbol ulaza ne menja

Primer rada SR parsera

1. $\langle S \rangle \rightarrow \langle A \rangle \langle A \rangle \langle S \rangle$

2. $\langle S \rangle \rightarrow \varepsilon$

3. $\langle S \rangle \rightarrow a$

4. $\langle A \rangle \rightarrow a$

	$\langle S \rangle$	$\langle A \rangle$	a
∇	$\langle S \rangle_0$	$\langle A \rangle_{11}$	a_x
$\langle S \rangle_0$			
$\langle A \rangle_{11}$		$\langle A \rangle_{12}$	a_4
a_x			
$\langle A \rangle_{12}$	$\langle S \rangle_1$	$\langle A \rangle_{11}$	a_x
a_4			
$\langle S \rangle_1$			

	a	$- $
∇	SHIFT	REDUCE (2)
$\langle S \rangle_0$	REJECT	ACCEPT
$\langle A \rangle_{11}$	SHIFT	REJECT
a_x	REDUCE (4)	REDUCE (3)
$\langle A \rangle_{12}$	SHIFT	REDUCE (2)
a_4	REDUCE (4)	REDUCE (4)
$\langle S \rangle_1$	REJECT	REDUCE (1)

sadržaj steka

1. ∇
2. ∇a_x
3. $\nabla \langle A \rangle_{11}$
4. $\nabla \langle A \rangle_{11} a_4$
5. $\nabla \langle A \rangle_{11} \langle A \rangle_{12}$
6. $\nabla \langle A \rangle_{11} \langle A \rangle_{12} \langle S \rangle_1$
7. $\nabla \langle S \rangle_0$

preost. ulaz akcija parsera

- $aa-|$ SHIFT
- $a-|$ REDUCE (4)
- $a-|$ SHIFT
- $-|$ REDUCE (4)
- $-|$ REDUCE (2)
- $-|$ REDUCE (1)
- $-|$ ACCEPT



Parsiranje od dna ka vrhu (BU)

Koncept BU parsiranja

- Kod pristupa 'odozdo-nagore' parser otkriva strukturu **stabla izvođenja** počev **od listova stabla nagore**, određujući čvorove roditelja prepoznavanjem leve strane smena na osnovu poznate desne strane.
- redosled smena koje parser prepoznaje tokom rada **obrnut** od redosleda pri krajnje **desnom izvođenju** date sentence



Rad potisnog BU parsera

- u svakom koraku važi:
stek + ulaz = sentencijalna forma u
krajnje desnom izvođenju ulaza iz $\langle S \rangle$
- parser potiskuje (akcija **SHIFT**) simbole
sa ulaza na stek dok ne kompletira
ručku
 - **SHIFT** \equiv **PUSH**(tekući ulazni simbol),
ADVANCE
- tada se ručka zamenjuje levom stranom
smene (akcija **REDUCE**)

Primer...

$$\langle S \rangle \Rightarrow_{rm} \langle A \rangle \langle A \rangle \langle S \rangle \Rightarrow_{rm} \langle A \rangle \langle A \rangle \Rightarrow_{rm} \langle A \rangle a \Rightarrow_{rm} a a$$

1

2

4

4

sadržaj steka

preost. ulaz akcija parsera

1. $\langle S \rangle \rightarrow \langle A \rangle \langle A \rangle \langle S \rangle$

1. ∇

aa-| **SHIFT**

2. $\langle S \rangle \rightarrow \varepsilon$

2. ∇a_x

a-| **REDUCE (4)**

3. $\langle S \rangle \rightarrow a$

3. $\nabla \langle A \rangle_{11}$

a-| **SHIFT**

4. $\langle A \rangle \rightarrow a$

4. $\nabla \langle A \rangle_{11} a_4$

-| **REDUCE (4)**

5. $\nabla \langle A \rangle_{11} \langle A \rangle_{12}$

-| **REDUCE (2)**

6. $\nabla \langle A \rangle_{11} \langle A \rangle_{12} \langle S \rangle_1$

-| **REDUCE (1)**

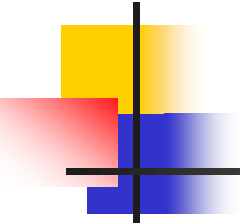
7. $\nabla \langle S \rangle_0$

-| **ACCEPT**



Pojam ručke (handle)

- **Ručka** sentencijalne forme α je pojava, u toj sentencijalnoj formi, **desne strane** produkcije koja je **poslednja primenjena** u krajnje desnom izvođenju α .
- **Produkcija ručke** je poslednja produkcija primenjena u krajnje desnom izvođenju sentencijalne forme.
 - *u prethodnom primeru, ručka za finalnu sentencu je početno slovo d, a produkcija ručke je 4.*

- 
-
- Uvod
 - SHIFT-REDUCE Parsiranje
 - Konstrukcija LR(0) parsera
 - Konflikti u SHIFT-REDUCE parsiranju



Konstrukcija SR parsera za LR(0) gramatike

- strategija odlučivanja se zasniva isključivo na sadržaju steka.
- Ulazni simboli ne uzimaju se u obzir pri odlučivanju.
- Značenje **LR(0)**:
 - ulaz se čita s Leva na desno;
 - nalazi se desno (Rightmost) izvođenje
 - gleda se 0 simbola na ulazu



Pojam LR(0) konfiguracije (engl. configuration ili item)

- LR(0) konfiguracija predstavlja gramatičku smenu sa dodatom tačkom negde na desnoj strani
- $A \rightarrow X_1 \dots X_i \bullet X_{i+1} \dots X_n$



Primer

■ za gramatiku:

1. $\langle S \rangle \rightarrow a$

2. $\langle S \rangle \rightarrow (\langle S \rangle \langle R \rangle$

3. $\langle R \rangle \rightarrow , \langle S \rangle \langle R \rangle$

4. $\langle R \rangle \rightarrow)$

LR(0) konfiguracije 2. smene

- konfiguracija zatvaranja (tačka na početku)
 $\langle S \rangle \rightarrow \bullet (\langle S \rangle \langle R \rangle$
- bazične konfiguracije (tačka nije na početku):
 $\langle S \rangle \rightarrow (\bullet \langle S \rangle \langle R \rangle$
 $\langle S \rangle \rightarrow (\langle S \rangle \bullet \langle R \rangle$
 $\langle S \rangle \rightarrow (\langle S \rangle \langle R \rangle \bullet$
- druga oznaka za poslednju konfiguraciju
 $\langle R \rangle_3$



Primer

- za praznu smenu $\langle X \rangle \rightarrow \varepsilon$ jedna konfiguracija, u oznaci
 $\langle X \rangle \rightarrow \varepsilon \bullet$ ili $\langle X \rangle \rightarrow \bullet$
- proširenje gramatike (uniformiše pravila konstrukcije parsera)
 0. $\langle S' \rangle \rightarrow \langle S \rangle \dashv$

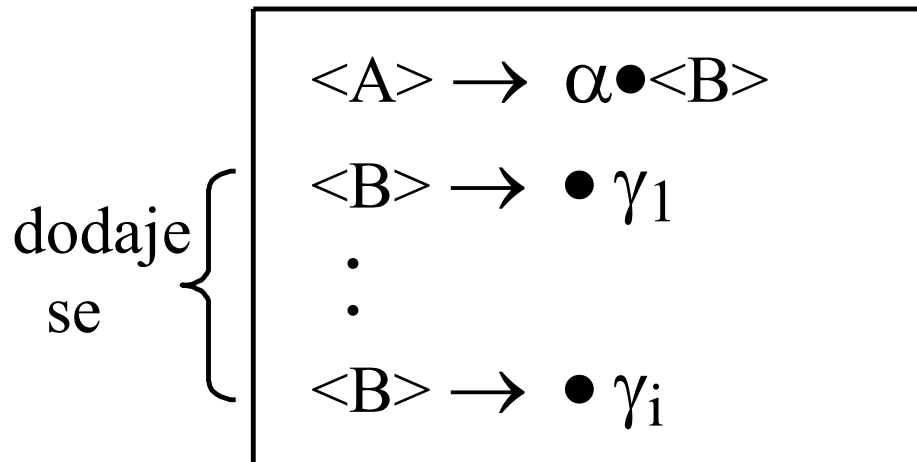


Konstrukcija karakterističnog LR(0) automata – detektora ručki

- Stanja automata su skupovi konfiguracija
- Iterativna konstrukcija grafa prelaza koristeći operacije **Closure0** i **Goto0**

Konstrukcija karakterističnog LR(0) automata – detektora ručki

- Operacija **Closure0** nad skupom konfiguracija s :
- $\text{Closure0}(s) = s \cup \{ \langle B \rangle \rightarrow \bullet \gamma \mid \langle A \rangle \rightarrow \alpha \bullet \langle B \rangle \beta \in \text{Closure0}(s) \}$

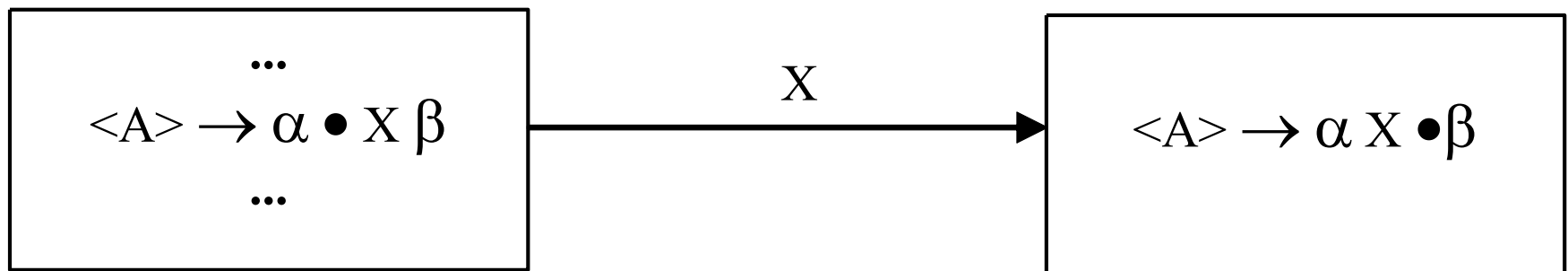


iterativno

$$\begin{array}{l} \langle A \rangle \rightarrow \bullet \langle B \rangle \dots \\ \langle B \rangle \rightarrow \bullet \langle C \rangle \dots \\ \langle C \rangle \rightarrow \bullet \dots \end{array}$$

Konstrukcija karakterističnog LR(0) automata – detektora ručki

- Operacija Goto0 nad stanjem s i ulaznim simbolom X :
- $\text{Goto0}(s, X) = \text{Closure0}(\{ \langle A \rangle \rightarrow \alpha X \bullet \beta \mid \langle A \rangle \rightarrow \alpha \bullet X \beta \in s \})$





Algoritam konstrukcije karakterističnog LR(0) automata

- Startno stanje detektora ručki se dobija kao $\text{Closure0}(\{ \langle S' \rangle \rightarrow \bullet \langle S \rangle \mid \})$.
- Stanje S' u koje se prelazi iz nekog stanja S pod ulazom X , dobija se kao rezultat procedure $\text{GoTo0}(S, X)$.
- Polazeći od startnog stanja, sistematski se određuju prelazi za svaki ulazni simbol, što generiše nova stanja.
- Ova stanja se procesiraju na isti način kao startno, sve dok postoji neko neobrađeno stanje.

Karakteristični automat za gramatiku iz primera



$\langle S' \rangle \rightarrow * \langle S \rangle \text{ --|}$

Karakteristični automat za gramatiku iz primera



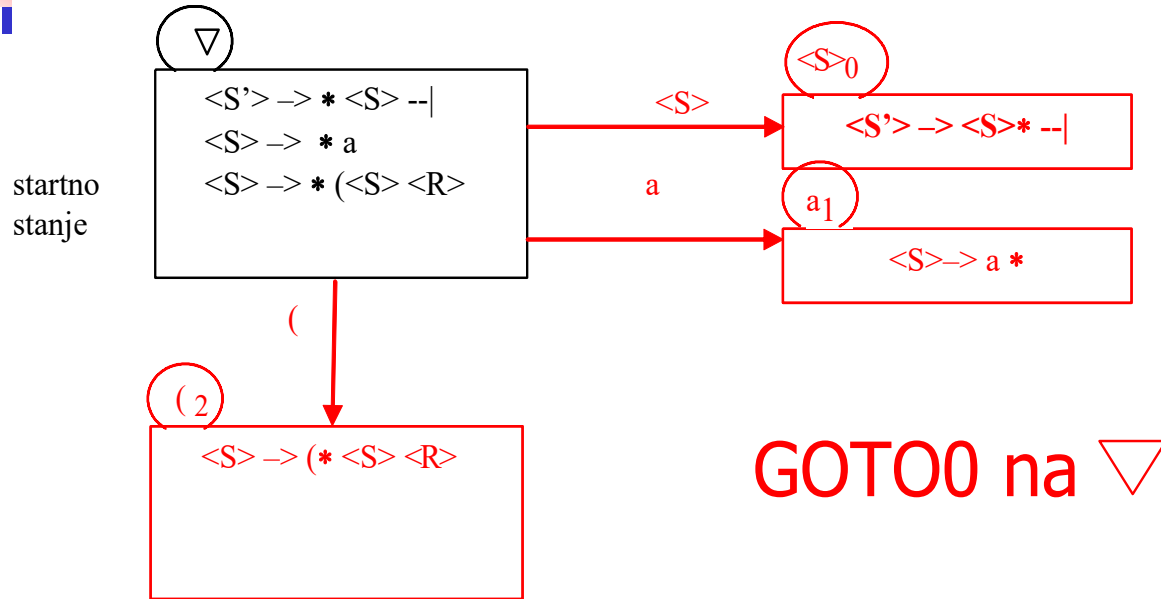
$\langle S' \rangle \rightarrow * \langle S \rangle \text{ --|}$

$\langle S \rangle \rightarrow * a$

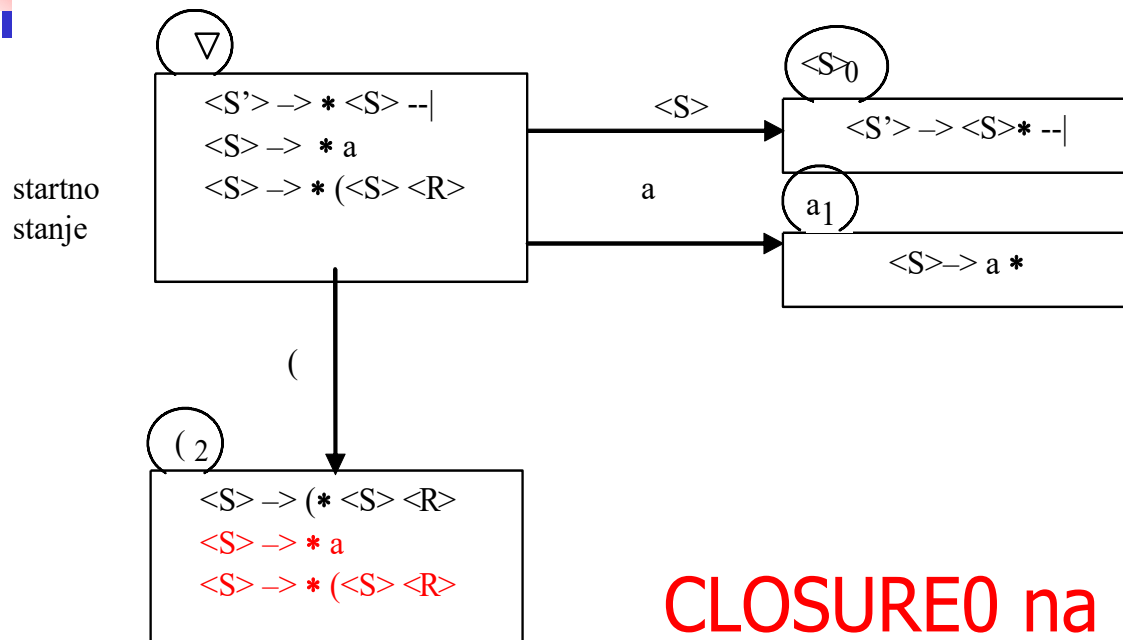
$\langle S \rangle \rightarrow * (\langle S \rangle \langle R \rangle)$

Closure0

Karakteristični automat za gramatiku iz primera

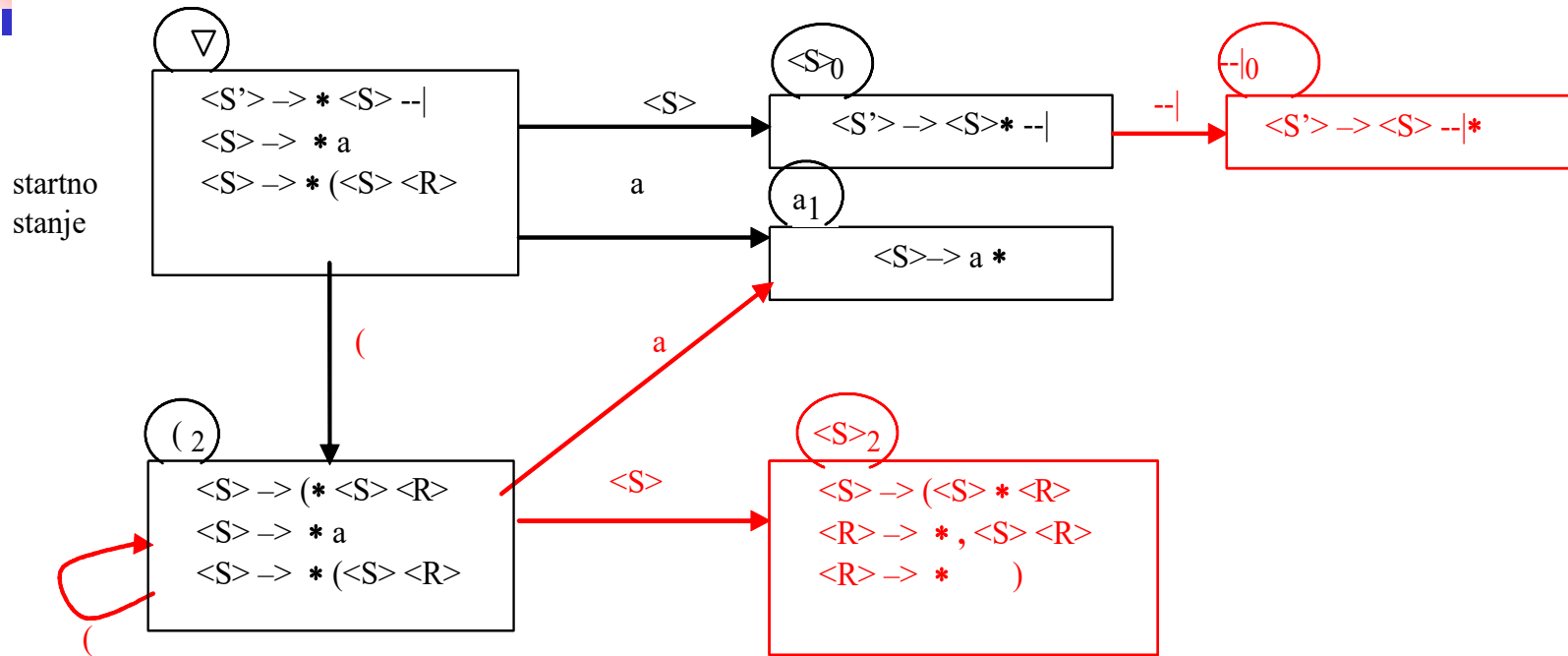


Karakteristični automat za gramatiku iz primera



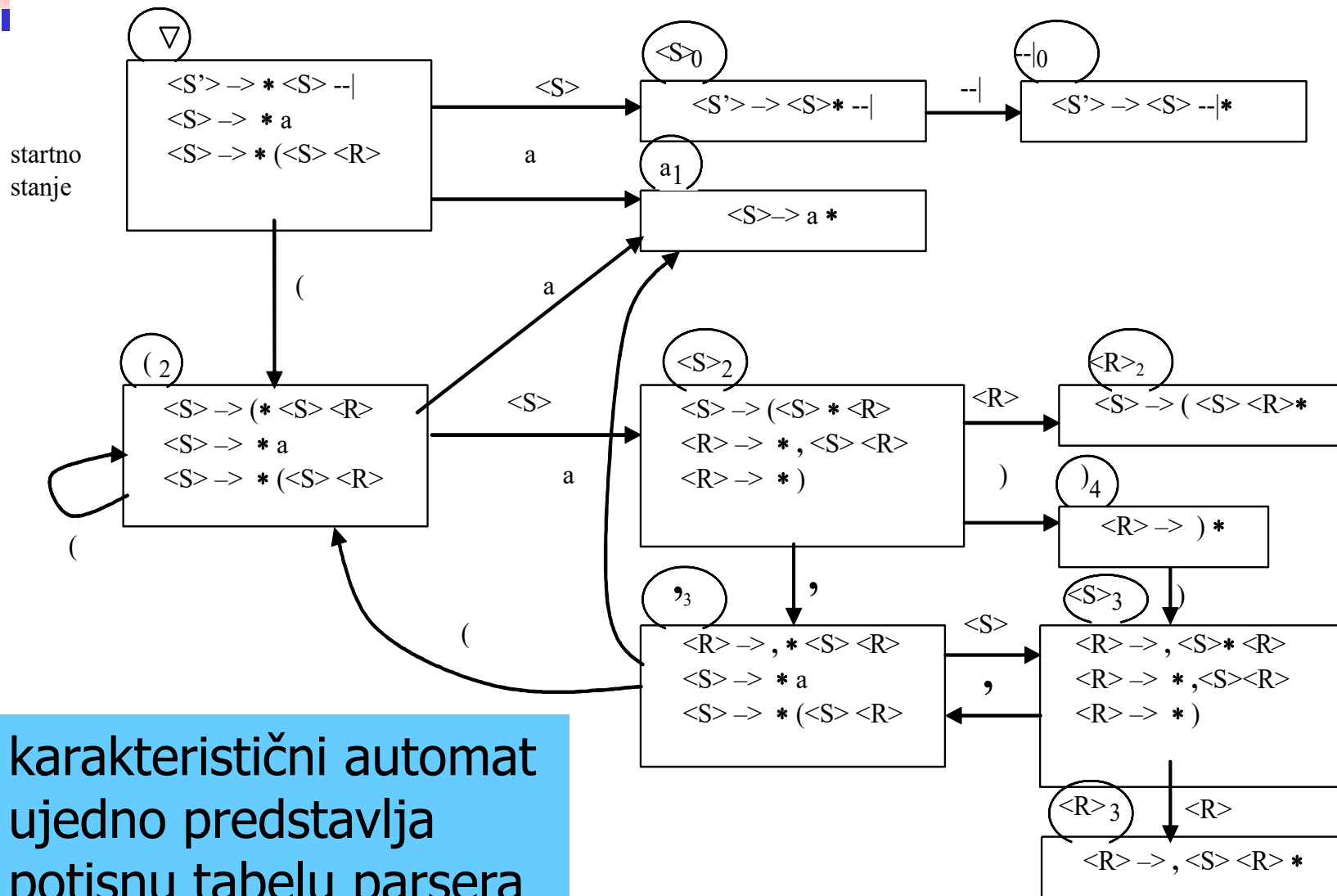
CLOSURE0 na nova stanja

Karakteristični automat za gramatiku iz primera



GOTO0 na nova stanja

Karakteristični automat za gramatiku iz primera



karakteristični automat
ujedno predstavlja
potisnu tabelu parsera

Popunjavanje LR(0) kontrolne tabele na osnovu karakterističnog automata

(razmatra se ulaz u vrsti V , kojoj odgovara skup konfiguracija S):

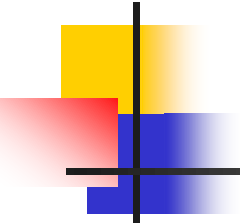
- Ako S sadrži konfiguraciju oblika $\langle X \rangle \rightarrow \alpha \bullet x \beta$, ulaz treba popuniti akcijom **SHIFT**.
- Ako S sadrži konfiguraciju oblika $\langle Y \rangle \rightarrow \gamma \bullet$, odnosno tačka se nalazi na kraju desne strane, ulaz treba popuniti akcijom **REDUCE**($\langle Y \rangle \rightarrow \gamma$). Izuzetno, ako se radi o nultoj smeni, akcija je **ACCEPT**.
- Vrste koje nisu pokrivene ni 1. ni 2. pravilom treba da sadrže akciju **REJECT**.
- Ukoliko gornja pravila na jednoznačan način određuju akciju za svaki ulaz kontrolne tabele, gramatika pripada klasi LR(0).

Primer

	$\langle S \rangle$	$\langle R \rangle$	(,)	a	\mid	
∇	$\langle S \rangle_0$		(₂			a ₁		SHIFT
$\langle S \rangle_0$							\mid_0	SHIFT
\mid_0								ACCEPT
a ₁								REDUCE(1)
(₂	$\langle S \rangle_2$		(₂			a ₁		SHIFT
$\langle S \rangle_2$		$\langle R \rangle_2$, ₃) ₄			SHIFT
$\langle R \rangle_2$								REDUCE(2)
, ₃	$\langle S \rangle_3$		(₂			a ₁		SHIFT
$\langle S \rangle_3$		$\langle R \rangle_3$, ₃) ₄			SHIFT
$\langle R \rangle_3$								REDUCE(3)
) ₄								REDUCE(4)

potisna tabela

kontrolna tabela

- 
-
- Uvod
 - BU parser u vidu potisnog automata
 - SHIFT-REDUCE Parsiranje
 - Konstrukcija LR(0) parsera
 - Konflikti u SHIFT-REDUCE parsiranju



Konflikti u SR parsiranju

- pri konstrukciji tabele mogu se pojaviti konflikti:
 - SHIFT/REDUCE, na primer, ako stanje sadrži
$$\langle X \rangle \rightarrow a \bullet$$
$$\langle Y \rangle \rightarrow a \bullet b$$
 - REDUCE/REDUCE, na primer, ako stanje sadrži
$$\langle X \rangle \rightarrow c \bullet$$
$$\langle Y \rangle \rightarrow d \bullet$$



Konflikti u SR parsiranju

- prazne smene uvek izazivaju konflikt
=> LR(0) ne dozvoljava prazne smene
- $\langle S' \rangle \rightarrow \bullet \langle S \rangle \text{ —|}$
- $\langle S \rangle \rightarrow \bullet a$
- $\langle S \rangle \rightarrow \bullet$



Konflikti u SR parsiranju

- tada gramatika nije LR(0) i (teorijski) nije moguće napraviti LR(0) parser;
- u praksi (yacc):
 - S/R konflikt uvek razrešava u korist S, osim ako se %prec direktivom eksplicitno ne zada drugačije
 - R/R konflikt (ozbiljniji problem) prema redosledu navođenja smena